RESEARCH ARTICLE OPEN ACCESS

An Improved Color Image Compression Approach based on Run Length Encoding

Uqba bn Naffa
Computer Science Department, University of Mustansiriyah, Baghdad, Iraq

Abstract:

Image compression is the noticeable requirement of recent digital image processing strategies as well as codes to save large digital images in small images. And for the same reason we need the image compression algorithms which has optimum performance of compression without losing visual quality of image. This paper presents an improved color image compression approach that has the ability to compress the color image. The proposed approach divides the color image into RGB bands; each band is selected to be divided. The division processes of the band into blocks are based on specific criteria with non-overlapping. For each selection, the some operations are applied. A particular implementation of this approach was tested, and its performance was quantified using the peak signal-to-noise ratio and similarity index. Numerical results indicated general improvements in visual quality for color image coding.

Keywords — **Image compression, Color image, DCT transforms, Image coding**.

I. INTRODUCTION

Compression is any method reducing the original amount of data to another less quantity. One can categorize already elaborated algorithms to lossless or lossy techniques: In the first case: the quality is totally preserved and we converse about storage or transmission reduction. In the counterpart, lossy algorithms look for the check of the well-known tradeoff rate-distortion. It means that the quality of the decompressed data must be in the tolerable bounds defined by each specific application according to the maximum possible compression ratio reachable [1].

In the recent years there has been an astronomical increase in the usage of computers for a variety of tasks. One of the most common usage has been the storage, manipulation, and transfer of digital images. The files that comprise these images, however, can be quite large and can quickly take up precious memory space on the computer's hard drive. In multimedia application, most of the images are in color. The color images contain lot of data redundancy and require a large amount of storage space. Image compression refers to the reduction of the size of the data that images contain. Generally, image compression schemes in [2, 3] exploit certain data redundancies to convert the image to a smaller form.

For compression, a luminance-chrominance representation is considered superior to the RGB representation. Therefore, RGB images are transformed to one of the luminance-chrominance models, performing the compression process, and then transform back to RGB model because displays are most often provided output image with direct RGB model. The luminance component represents the intensity of the image and look like a gray scale version. The chrominance components represent the color information in the image [4]. Douak et al. [5] have proposed a new algorithm for color images compression.

Mohamed et al. [6] proposed a hybrid image compression method, which the background of the image is compressed using lossy compression and the rest of the image is compressed using lossless compression. In hybrid compression of color images with larger trivial background by histogram segmentation, input color image is subjected to binary segmentation using histogram to detect the background. The color image is compressed by

standard lossy compression method. The difference between the lossy image and the original image is computed and is called as residue. The residue at the background area is dropped and rest of the area is compressed by standard lossless compression method. This method gives lower bit rate than the lossless compression methods and is well suited to any color image with larger trivial background.

II. ARCHITECTURE FOR THE COMPRESSION TECHNIQUE STANDARD

The compression technique contains the four "modes of operation". For each mode, one or more distinct codec's are specified. Codec's within a mode differ according to the precision of source image samples they can handle or the entropy coding method they use. Although the word codec (encoder/decoder) is used frequently in this project, there is no requirement that implementations must include both an encoder and a decoder. Many applications will have systems or devices which require only one or the other [7].

The four modes of operation and their various codec's have resulted from JPEG's goal of being generic and from the diversity of image formats across applications. The multiple pieces can give the impression of undesirable complexity, but they should actually be regarded as a comprehensive "toolkit" which can span a wide range of continuous-tone image applications. It is unlikely that many implementations will utilize every tool -- indeed, most of the early implementations now on the market (even before final ISO approval) have implemented only the Baseline sequential codec. Figures (1) and (2) show the key processing steps which are the heart of the DCT-based modes of operation. These figures illustrate the special case of single-component (gray scale) image compression [8].

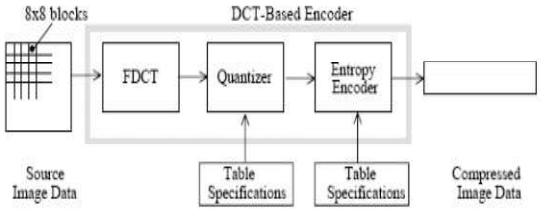


Fig. 1 DCT-Based Encoder Processing Steps

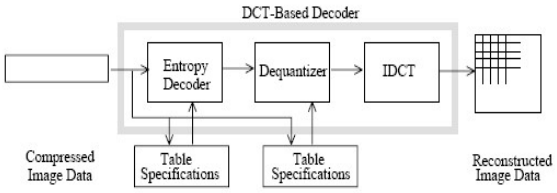


Fig. 2 DCT-Based Decoder Processing Steps

A. The 8x8 FDCT and IDCT

At the input to the encoder, source image samples are grouped into 8x8 blocks, and a DCT is performed on each block, processing them from left to right, top to bottom, and input to the Forward DCT (FDCT). At the output from the decoder, the Inverse DCT (IDCT) outputs 8x8 sample blocks to form the reconstructed image. The following equations are the idealized mathematical definitions of the 8x8 FDCT and 8x8 IDCT [8]:

$$FDCT(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}p(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$IDCT(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)DCT(u,v)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$
$$\approx p(x,y)$$

$$C(u,v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u,v = 0 \\ 1 & \text{if } u,v > 0. \end{cases}$$

Where, the JPEG compression algorithm works in three steps: the image is first transformed, and then the coefficients are quantized, and finally encoded with a variable-length lossless code.

B. Quantization

After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a 64-element Quantization Table, which must be specified by the application (or user) as an input to the encoder. Each element can be any integer value from 1 to 255, which specifies the step size of the quantizer for its corresponding DCT coefficient. The purpose of quantization is to achieve further compression by representing DCT coefficients with

no greater precision than is necessary to achieve the desired image quality. Stated another way, the goal of this processing step is to discard information which is not visually significant. Quantization is a many-to-one mapping, and therefore is fundamentally lossy. It is the principal source of lossless in DCT-based encoders.

Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer [8]:

$$F^{Q}(u,v) = Integer\ Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

This output value is normalized by the quantizer step size. Dequantization is the inverse function, which in this case means simply that the normalization is removed by multiplying by the step size, which returns the result to a representation appropriate for input to the IDCT:

$$F^{Q'}(u,v) = F^{Q}(u,v) * Q(u,v)$$

The quantized values are then reordered according to a zigzag pattern shown in Figures (3) and (4).

DC → Horizontal Frequency →

Vertical Frequency ↓

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig. 3 The Quantization Table.

To take advantage of the slow varying nature of most natural images, the DC coefficient is predicted from the DC coefficient of the previous block and differentially encoded with a variable length code.

The rest of the coefficients (referred to as AC coefficients) are run-length coded. JPEG produces relatively good performance at medium to high coding rates, but suffers from blocking artifacts at lower rates because of the block-based encoding.

At low rates, in order to meet the target coding rate, the quantization table chosen is so coarse that only the DC coefficient is encoded without the higher frequency details. At the decoder this creates visually noticeable discontinuities at the block boundaries [7].

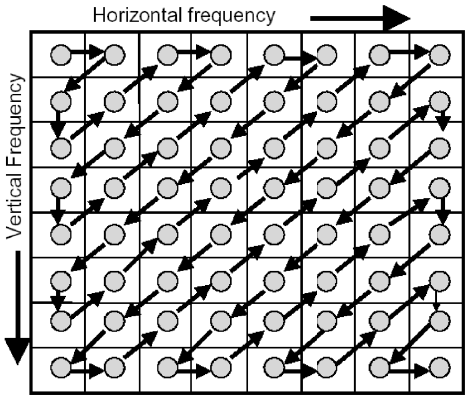


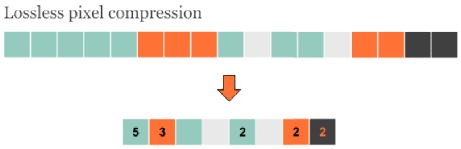
Fig 4. Zigzag ordering of coefficients of the 8 x 8 blocks in the JPEG algorithm.

C. Entropy Coding

The final DCT-based encoder processing step is entropy coding. This step achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. The JPEG proposal specifies two entropy coding methods Huffman coding [7] and Run Length coding [5].

D. Run-length encoding

Run-length encoding is a data compression algorithm that helps us encode large runs of repeating items by only sending one item from the run and a counter showing how many times this item is repeated. Unfortunately this technique is useless when trying to compress natural language texts, because they don't have long runs of repeating elements. In the other hand RLE is useful when it comes to image compression, because images happen to have long runs pixels with identical color. As you can see on the following picture we can compress consecutive pixels by only replacing each run with one pixel from it and a counter showing how many items it contains [9].



III. The Proposed Image Compression Approach

In order to obtain the best possible compression ratio (CR), Discrete cosine transform (DCT) has been widely used in image and video coding systems, where zigzag scan is usually employed for DCT coefficient organization and it is the last stage of processing a compressed image in a transform coder, before it is fed to final entropy encoding stage. The basic idea of the new approach is to divide the image into 8×8 blocks and then extract the consecutive non-zero coefficients preceding the zero coefficients in each block. The decompression process can be performed systematically and the number of zero coefficients can be computed by subtracting the number of non-zero coefficients from 64 for each block. The block diagram of proposed image compression approach is shown in Fig.5.

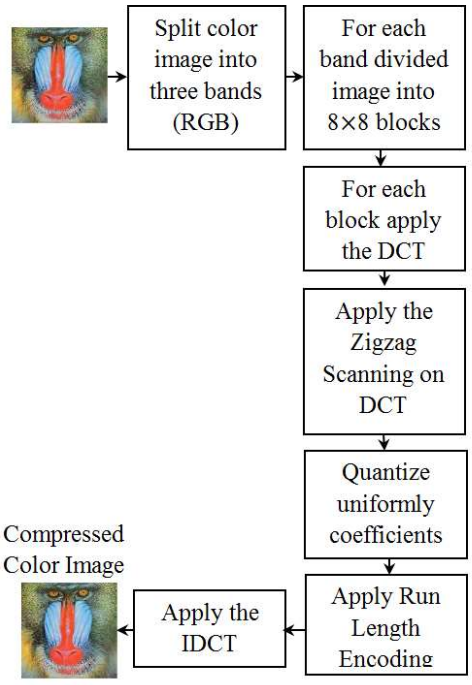


Fig 5.The block diagram of proposed image compression approach

IV. Experimental Results

For the implementation and evaluation of the approach we developed a visual basic 6 code and performed the testing on a standard color test images Lena, Fruit and Airplane of size 256×256 (see Figure 6). We analyze the results obtained with the first, second and finally, the proposed algorithm. All images and tables from the experiments are given. Standard measures for image compression [9], like compression ratio (CR) and peak signal to noise ratio (PSNR) and structural similarity (SSIM) were used, which are calculated for comparing the performance of the proposed approach as per the following representations:

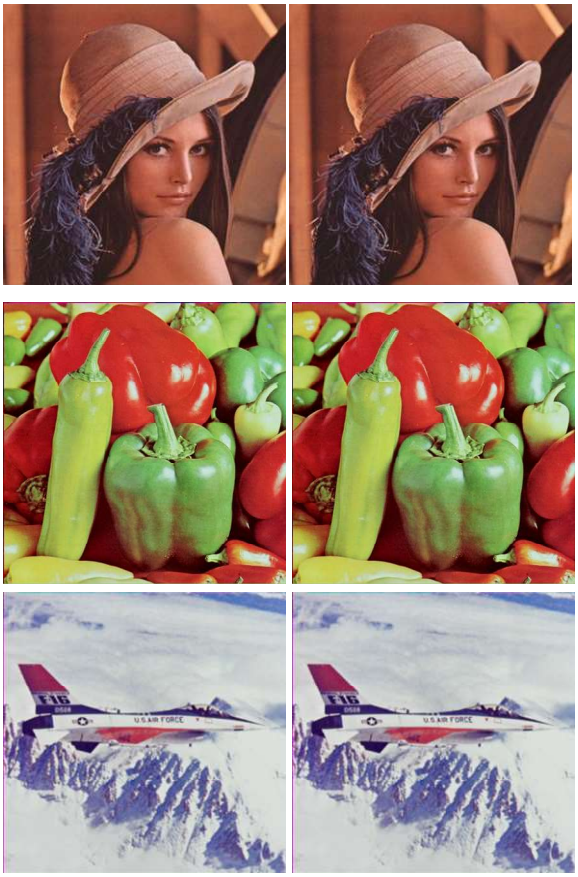


Fig 6. The results of proposed approach, left column represent the original images, right column represent the compressed images.

$$CR = \frac{orginal\ image\ size\ in\ bytes}{compressed\ image\ size\ in\ bytes}$$

From the results listed in Table (1), the proposed codec achieves a high performance.

TABLE I
NUMERICAL RESULTS FOR PROPOSED METHOD.

Image	PSNR	SSIM	CR
Lena	44.67	0.903	10.5
Fruit	42.32	0.879	16.7
Airplane	43.88	0.894	14.5

V. CONCLUSIONS

In this paper an improved color image compression approach was proposed along with its applications to compress color images. The obtained results shows the improvement of the proposed method over the recent published paper both in quantitative PSNR terms and very particularly, in visual quality of the reconstructed images. Furthermore, it increased the compression rate.

REFERENCES

[1]. Pallavi N. Save and Vishakha Kelkar, "An Improved Image Compression Method using LBG with DCT", IJERT Journal, Volume-3,Issue-06, June-2014.

[2]. X. O. Zhao and Z. H. He, "Lossless image compression using super-spatial structure prediction," Signal Processing Letters, IEEE, vol. 17, no. 4, pp. 383–386, 2010.

[3]. W.M. Abd-Elhafiez, "Image compression algorithm using a fast curvelet transform," International Journal of Computer Science and Telecommunications, vol. 3, no. 4, pp. 43–47, 2012.

[4]. M. Sonka, V. Halva, and T.Boyle, "Image Processing Analysis and Machine Vision", Brooks/Cole Publishing Company, 2nd Ed., 1999.

[5]. F. Douak, Redha Benzid, Nabil Benoudjit "Color image compression algorithm based on the DCT transform combined to an adaptive block scanning," Int. J. Electron. Commun. (AEU), vol. 65, pp. 16–26, 2011.

[6]. M. Mohamed Sathik, K.Senthamarai Kannan and Y.Jacob Vetha Raj, "Hybrid Compression of Color Images with LargerTrivial Background by Histogram Segmentation", (IJCSIS) International Journal of Compute.

[7]. Makrogiannis S, Economou G, Fotopoulos S. Region oriented compression of color images using fuzzy inference and fast merging," Pattern Recognition, 35:1807–20, 2002.

[8]. Alkholidi A, Alfalou A, Hamam H. "A new approach for optical colored image compression using the jpeg standards," Signal Processing, 87:569–83, 2007.

[9]. G. Sreelekha, P.S. Sathidevi, "An HVS based adaptive quantization scheme for the compression of color images", Digital Signal Processing, Vol. 20(4), pp. 1129-1149, 2010.